WHAT IS CLAIMED IS:

SUB A17

~~1. A method for processing a plurality of files to create a single, executable file, comprising:~~

~~creating a single output file;~~

~~copying executable code to the output file;~~

~~writing destination information to the output file to designate the destination directory of the executable file;~~

~~writing plural blocks of data to the output file, each block containing identification information and corresponding data;~~

~~writing a block containing a clean-up program to the output file if the destination information corresponds to a temporary file; and~~

~~writing auto-start file information to the output file to designate a file to be opened when the output file is executed, if an auto-start file is specified by an author.~~

~~2. The method of claim 1, wherein writing plural blocks comprises writing the corresponding data in a compressed format.~~

~~3. The method of claim 1, wherein writing the blocks comprises writing a block start flag for each block.~~

~~4. The method of claim 1, further including receiving user input to identify the destination directory.~~

5. The method of claim 1, further including writing a source-identifying block to the output file to indicate the source of the file.

6. The method of claim 1, further including:

running the executable code to identify one of the blocks;

processing identification information contained in the block to determine the contents of the block;

reading the data in the block and creating a corresponding directory if the block is a destination directory block;

decompressing the data in the block and writing the decompressed data to an appropriate directory if the block is a compressed file block;

writing the data in the block to a temporary directory if the block contains a clean-up program; and

saving the information in the block if the information contains auto-start path information.

7. The method of claim 6, further including:

beginning a display of data at a preselected position;

determining a current position of the display;

comparing the determined position with a set of event data for the respective digital assets;

displaying one of the digital assets based on the comparison of the position with the event data;

calculating a timeout based on the determined position and the event data;

setting a clock to fire upon reaching the timeout;

initiating a polling process when the clock fires to determine the position of the display;

displaying a different digital asset based on a comparison of the determined position with the event data; and

calculating a new timeout and resetting the clock to fire upon reaching the new timeout.

8. The method of claim 6, wherein reading the data further comprises determining whether the data corresponds to a temporary directory, and creating an entry to execute the clean-up program if the data corresponds to a temporary directory.

9. The method of claim 6, further including determining whether the clean-up program is needed, and writing the clean-up program to the temporary directory only if it is needed.

10. The method of claim 6, further including determining, after the blocks have been written to the appropriate destinations, if an auto-start file is specified, and opening the auto-start file if it is specified.

11. The method of claim 6, further including processing a source-identifying block to verify the source of the executable file.

~~12. A method of unpackaging and launching an executable file, comprising:~~

~~providing the executable file including executable code and a plurality of blocks of data;~~

~~running the executable code to identify one of the blocks;~~

~~processing identification information contained in the block to determine the contents of the block;~~

~~reading the data in the block and creating a corresponding directory if the block is a destination directory block;~~

~~decompressing the data in the block and writing the decompressed data to an appropriate directory if the block is a compressed file block;~~

~~writing the data in the block to a temporary directory if the block contains a clean-up program; and~~

~~saving the information in the block if the information contains auto-start path information.~~

~~13. The method of claim 12, wherein reading the data further comprises determining whether the data corresponds to a temporary directory, and creating an entry to execute the clean-up program if the data corresponds to a temporary directory.~~

~~14. The method of claim 12, further including determining whether the clean-up program is needed, and writing the clean-up program to the temporary directory only if it is needed.~~

~~15. The method of claim 12, further including determining, after the blocks have been written to the appropriate destinations, if an auto-start file is specified, and opening the auto-start file if it is specified.~~

~~16. The method of claim 12, further including processing a source-identifying block to verify the source of the executable file.~~

ADD A27